2004 Society for Design and Process Science
Printed in the United States of America

ASSESSMENT OF COUPLING AND COHESION FOR COMPONENT-BASED SOFTWARE BY USING SHANNON LANGUAGES

R. Seker
Computer Science Department, Texas Tech University, Lubbock, Texas, USA

A. J. van der Merwe
P. Kotze
Department of Computing, University of South Africa, Pretoria, South Africa

M. M. Tanik
Electrical and Computer Engineering Department, University of Alabama at Birmingham, Birmingham, Alabama, USA

R. Paul
Department of Defense, USA

Component-Based Software (CBS) engineering is envisioned to address the issues related to the increasing size and complexity of software systems. In CBS development, the designer designs systems by using readily available (possibly third party) software components without needing the source code for the components. Lack of source code, in general, renders the classical metrics cumbersome to use, if not useless. Coupling and cohesion aspects of a system/subsystem are the quality attributes that can seriously impact the maintenance, evolution, and reuse. We present an information-theoretic approach based on the notion of Shannon Languages for helping the system designer in the assessment of coupling and cohesion early in the design phase. The proposed methodology is most beneficial for CBS (where the source code is in general absent) however it is applicable in other development methodologies in which the source code for the software components is available.

Keywords: *component-based software, Shannon languages, coupling, cohesion*

1. Introduction

Currently the trend is that software systems are growing in size and complexity. In order to be able to develop large-scale software systems in an efficient way, reuse of existing software becomes critical (Seker *et al.*, 2004, Seker, 2002, Sage and Palmer, 1990, Tanik and Chan, 1991, Jololian, 2000, Hopkins, 2000). Component-Based Software (CBS) engineering, envisioned by utilising high reusability, seems to be the software development methodology to alleviate the problems associated with building large-scale software systems (Wallnau *et al.*, 2002, Brereton and Budgen, 2000, Emmerich, 2002, Brown and Wallnau, 1998, Grundy *et al.*, 1998). CBS development (CBSD) practice is motivated by the accomplishments in manufacturing of electronic devices. Electronic devices are built from the components that have well-defined interfaces and functionalities. These components are integrated without knowledge of the actual design and circuit elements inside them (Cox and Baoming, 2001, Martins *et al.*, 2001). The analogy is that while building software systems from components, the system designer does not need the source code of the components. The designer only needs to know the components' functionalities and interfaces. CBSD therefore relies on the reuse of existing, well-defined components that are developed for integration (Jololian, 2000, Heineman and Councill, 2001). Building systems with components shifts the development niche from lines-of-code to coarser-grained components and their interconnections. CBSD focuses on the architecture, separates infrastructure from logic and helps in dealing with systems complexity. These attributes help easing development of the large-scale systems in a cost effective fashion (Brown and Wallnau, 1998).

Classical analysis and testing tools usually require the source code for the software system to be available. This poses a problem because the CBSD methodology is envisioned by not needing the source code for the components. Access to the source code becomes even harder when commercial off the shelf components are utilised. Even though the source code for components may be available, the components might have been implemented in different languages, environments, etc. This heterogeneity can pose additional difficulties for applying the classical measurement approaches to CBS.

A system designer faces a number of challenging issues during the design of the system in question. The components that make up the system and the relationships among them have to be identified. Furthermore, each component must have the maximum number of relationships within and minimum number of relationships with other components. These relationships are respectively referred to as cohesion and coupling. An ideal system possesses highly cohesive components which are loosely coupled to one another. Highly cohesive components exhibit high reusability and loosely coupled systems enable easy maintenance of the system due to the fact that a change made to a component is less likely to cause a regression fault in another component.

Although identification of components in a CBS system may be trivial, the relationships among components can impact the design decisions drastically. Coupling in a CBS system is a function of communication (relationship) between components. This viewpoint suggests that if two components are coupled, there is either one way or two way communication taking place between them. By focusing on the communication between components one can utilise the information-theoretic concepts for assessment of coupling in a CBS system. In this paper, we utilise information theory to assess the coupling in CBS systems. In order to make an assessment of cohesion in components, we take the same viewpoint; we view cohesion as a function of internal communication within a component. In other words, the intra-component communication (the communication taking place between the subcomponents of a component) is utilised for the assessment of cohesion in a component.

We utilise information theory for assessment of coupling and cohesion mainly because of unavailability source code for the components. Information theory has been utilised for the measurement and assessment of software systems (Seker *et al.*, 2004). Application of information theory to the software engineering domain has been mostly limited to entropy-based measures (Hellerman, 1972, Schutt, 1977, Coulter *et al.*, 1987, Allen and Khoshgoftaar, 1999). These works have mostly assumed a priori probability distributions for calculating the entropy.

Our information-theoretic approach to the assessment of CBS differs from the others in the sense that we utilise the capacity notion for the noiseless channels in information theory. The capacity of noiseless channels is also known as the capacity of Shannon languages.

Using graphs to represent programs enables the introduction of graph theoretical notions to solve problems in software engineering (Ramamoorthy, 1966, Muchnick and Jones, 1981, Ramamoorthy and Ho, 1975, Ramamoorthy *et al.*, 1976). One such approach is the investigation and derivation of software metrics using flowgraphs (van den Broek and van den Berg, 1995).

In this paper, we first consider strongly connected control flow graphs (CFGs) that represent the glue used to compose the components (system's design). T he nodes of a CFG represent components that make up the system, while the directed arcs represent the relationship between the components. The arc extending from one component's interface to the others may represent data flow, parameter passing, etc. Once we have the CFG representing the CBS system, there is a Shannon language defined on that particular CFG. We then utilise the capacity notion for the Shannon language defined on the CFG of interest. The capacity of the Shannon language defined on a CFG also corresponds to the combinatorial capacity of that CFG. We derive both the coupling and cohesion metrics from the combinatorial capacity of the CFG that represent the system or the component, respectively. The capacity of the Shannon language which is defined on the CFG of the CBS is calculated without assigning a priori probabilities. Not using the pre-assigned probabilities in applying information theory to the assessment of software is the distinctive approach presented by this paper. The paper concludes by giving a summary of results and stating some future research directions.

2. Methodology

A CBS system is composed of component integration units (CIUs) that are composed of components. A CIU can be referred to as a composite component, a subsystem, or a system. Throughout this paper, for the sake of simplicity, we will assume that the CIUs are composed in-house and are not received as third-party components. In other words, we know the CFG of the CIUs. The paper proceeds with providing the necessary background for the proposed methodology.

2.1. Coupling and Cohesion Metrics for CBS Systems

There exists extensive literature on Shannon's capacity notions for both noisy and noiseless channels (Shannon and Weaver, 1963, Khandekar *et al.*, 1999, Nambiar, 2001, Lind and Markus, 1999). In this paper, we focus on modelling CBS systems by noiseless channels and thereby assess coupling and cohesion in CBS systems. Despite the fact that information theory is a well-established field, it has found many application domains due to its generality. Some of the application areas for information theory that are relevant for this paper are investigation of randomness (Chaitin, 1975) and software engineering (Hellerman, 1972, Coulter *et al.*, 1987, Allen and Khoshgoftaar, 1999, Seker and Tanik, 2003a). In general, the application of information theory to software has been to define metrics through pre-assigned probabilities (usually equal probabilities). This paper differs from its counterparts in two ways:

(1) We utilise the capacity notion for noiseless channels rather than entropy

(2) after we model a CBS system with a CFG, the capacity and the proposed metrics are calculated via connectivity relationship between components rather than pre-assigned probabilities (for both nodes and arcs of graphs that represent the software system).

In the remainder of this section, we briefly provide the necessary background for the proposed methodology. The interested reader can refer to (Seker and Tanik, 2003b, Seker *et al.*, 2004, Seker and Tanik, 2002) for more thorough discussions on information-theoretic modelling of CBS systems and the details of calculations.

Calculating the combinatorial capacity of a Shannon language that is defined on a CFG requires utilising some graph-theoretical notions. The arcs of a CFG that represents a CBS system are labelled with a parameter that stands for connectivity and the theory of non-negative matrices is utilised for the calculation of capacity. We provide the necessary theorems and notions for calculating the capacity of Shannon languages by following (Shannon and Weaver, 1963, Khandekar *et al.*, 1999, Seker and Tanik, 2002).

Theorem 1 *Let A be the adjacency matrix of a graph G, G (A). A is irreducible if and only if G is strongly connected.*

Definition 1 *A channel that introduces zero noise uncertainty is a noiseless channel.*

Definition 2 *A Shannon language $L_{D,\varphi}$ is a language defined on a labelled directed graph D with a labelling φ.*

Definition 3 *Let s be a non-negative real number. The arc duration partition function for each pair of vertices (υ_i, υ_j), $P_{\upsilon_i,\upsilon_j}(s)$, is defined as*

$$P_{\upsilon_i,\upsilon_j}(s) = \sum_{b \in B\upsilon_i,\upsilon_j} e^{-s\tau_b} \tag{1}$$

where $B_{\upsilon_i,\upsilon_j}$ is the set of arcs directed from υ_i to υ_j, τ_b is the duration for arc b and $\upsilon \in V$, and V is the set of vertices (nodes) of the digraph. One can consider the arc duration partition functions $P_{\upsilon_i,\upsilon_j}(s)$ for each pair of vertices (υ_i, υ_j) as the (υ_i, υ_j) (i.e., (i, j)) entries of an $n \times n$ matrix P(s), which corresponds to the adjacency matrix of the digraph.

Definition 4 *$n \times n$ matrix P(s) is called the partition matrix.*

Theorem 2 *The combinatorial capacity of $L_{D,\varphi}$ is given by*

$$C_{comb} = s_0 \quad nats, \tag{2}$$

where s_0 is the unique solution to $\rho(s) = 1$ and $\rho(s)$ is the spectral radius of partition matrix P(s). Moreover, C_{comb} for $L_{D,\varphi}$, s_0 is the greatest positive solution of the equation q(s) = det(I - P (s)) = 0.

Change of a variable $x = e^{-s}$, eases the computations. The proposed change of variable enables us avoid having to solve the exponential equation $\rho(s) = 1$ but solve the polynomial $q(x) = \det(I - P(x)) = 0$, which is easier to solve. Instead of looking for the largest positive real root s0, we use the smallest positive real root x_0. Then, the combinatorial capacity C_{comb} becomes

$$C_{comb} = -\log_2 x_0 \quad bits/symbol \tag{3}$$

Existence of the desired root (smallest positive real root, the unique root) x_0 is guaranteed when the CFG is strongly connected (Minc, 1988, Berham and Plemmons, 1979, Lind and Markus, 1999, Seker and Tanik, 2001; 2002, Nambiar, 1996; 2001).

Capacity C_D of a Shannon language $L_{D,\varphi}$ is the maximum of the amount of average information (in nats or bits) generated at each node of graph *D*. When we set the durations for the arcs (τ_b) in the CFG to unity ($x^1 = x \Rightarrow \tau_b = 1$), the amount of information generated at a node corresponds to coupling or cohesion.

Definition 5 *Coupling between the components in a CBS system with m components is defined as*

$$\Upsilon_D = \frac{C_D}{C_{fc}} \times 100\% \tag{4}$$

C_{fc} is the capacity value calculated for the fully connected graph with m *nodes.*

Definition 6 *Cohesion of a component, represented by a directed graph D (with n nodes) is defined as*

$$\Omega_D = C_D,\ \textit{if component's "internals" are not known and}$$

$$\frac{C_D}{C_{fc}} \times 100\%,\ \textit{if component's "internals" are known.}$$

C_{fc} is the capacity value calculated for the fully connected graph with n *nodes.*

2.2. Application Examples

In this section, two examples for assessment of cohesion and coupling in a CBS systems are presented. Figure 1 shows four CIUs (components) each composed of four subcomponents as seen in (A) ... (D). The CIU in Figure 1 (A) is the least cohesive CIU since it is a CBS system which has a CFG that is strongly connected and has the minimum number of edges required for composing four subcomponents. A quick examination of the systems seen in this figure reveals that the number of relations (directed edges) increases as we move from the system in (A) towards the system in (D). The system in (D) is the maximally connected CFG with four subcomponents therefore it sets the maximum value for Ω_D a system with four components can have. Table 1 presents the cohesion metric Ω_D values as well as the capacity values for the Shannon languages defined on each of the CFGs (CBS systems) seen in Figure 1.

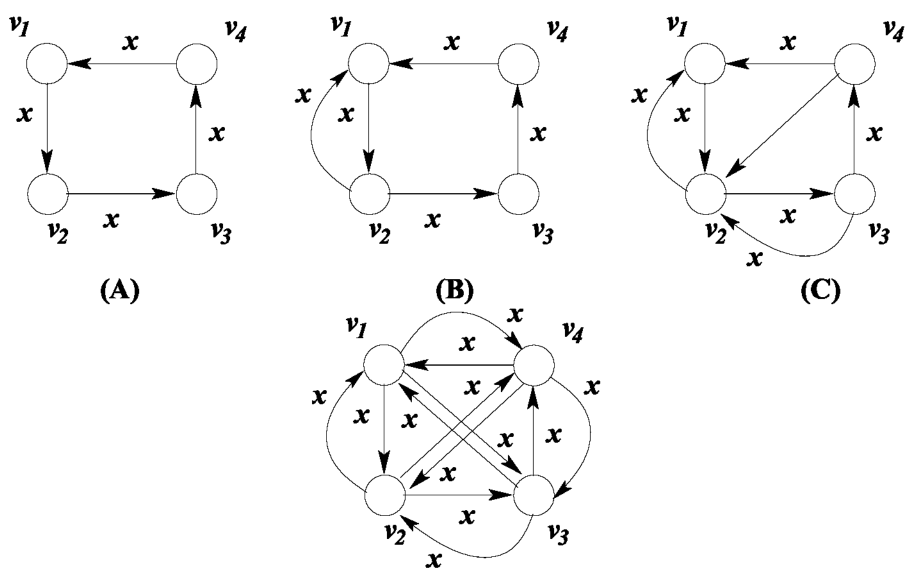

Fig. 1 A CBS system coupled in different ways.

The next example shows how the coupling metric Υ_D can be utilised for the assessment of coupling in a CBS system. Figure 2 presents two CIUs (far left in the figure) to be composed to make a CBS system. Each of these CIUs are composed of four components, hence each of the CBS systems in this figure has eight components. Figure 2 (A), (B) and (C) present the three CBS systems composed from the two CIUs seen in the far left of the figure. These two CIUs composed in different ways to make up different systems. The coupling metric values of these systems are given in Table 2. The capacity of the fully connected CFG with eight number of components is calculated to be C_{fc} = 2.80735492. By examining the CFG of the CBS systems in Figure 2 (A), (B) and (C), one can see that the coupling for the system in (C) is the smallest since some of the relations in the CIUs have been removed to compose this system. The system in (A) has higher coupling than the one in (C). The CBS system in (B) has a higher Υ_D value because of the specific components in the two CIUs at which the relationships are defined to make up the system (degrees of nodes at which the edges for composition arrive).

Table 1 Ω_D values for the CBS systems seen in Figure 1.

System	C_D	$\Omega_D = (C_D / C_{fc}) \times 100\%$
A	0	0
B	0.347120956	21.9008940
C	0.774539633	48.8680100
D	1.584962501	100

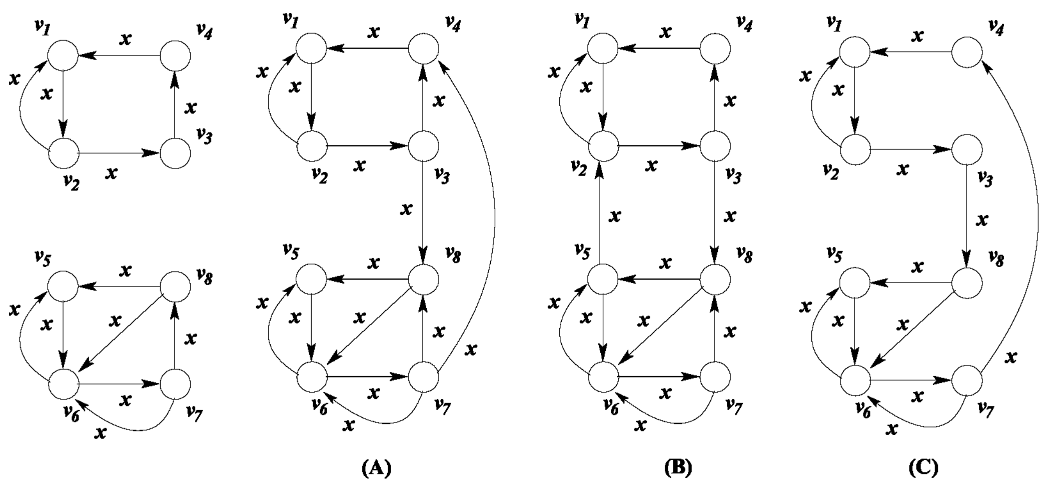

Fig. 2 A CBS system coupled in different ways.

Table 2 Υ_D values for the CBS systems seen in Figure 2.

System	C_D	$\Upsilon_D = (C_D / C_{fc}) \times 100\%$
A	0.8080887	28.7847010
B	0.8869796	31.5948512
C	0.6142308	21.8793426

2.3. Coupling (Υ_D) and Cohesion (Ω_D) Metrics versus Expected Properties

Briand *et al.* (1996) provide sets of properties that coupling and cohesion metrics should satisfy. The set of properties they have stated in their work is aimed for modular systems, which could be considered as CBS systems. However, our approach in this paper is that we do not have access to the source code of the components and in most cases, we do not know the components' internals.

It is recommended that a coupling or cohesion metric satisfy the properties proposed by (Briand *et al.*, 1996), hence we adapted these properties to be applicable in the domain of CBS systems where the source code is unavailable. We adapt their module definition to be a component in the CBS domain. A component that is composed from other components in-house will be referred to as a CIU, as discussed earlier. The properties to be satisfied by coupling and cohesion metrics for CBS systems are presented in Tables 3 and 4.

Table 3 Properties of coupling for CBS systems adapted from Briand *et al.*, 1996.

1.	Nonnegativity: Coupling of a CBS system must be nonnegative
2.	Null value: Coupling of a CBS system is null if there are no edges connecting components
3.	Monotonicity: Addition of edges between components in a CBS system doesn't decrease its coupling
4.	Composition of components: when two components Co_1 and Co_2 are composed into a CIU (new component) $Co_{1\cup 2}$, the coupling of the CBS containing the CIU $Co_{1\cup 2}$ is not greater than that of the original CBS system containing Co_1 and Co_2 –This property is valid as long as the composition information is not lost e. g. the "internals" of $Co_{1\cup 2}$ is known
5.	Disjoint component additivity: if two components Co_1 and Co_2 have no arc extending from one to the other, they are said to be disconnected. If these two components are combined into a new component (a new "CIU" with no arcs extending from one of the components to the other) $Co_{1\cup 2}$ (which replaces the components Co_1 and Co_2), the coupling of the CBS system doesn't change

Table 4 Properties of cohesion for CBS systems adapted from Briand *et al.*, 1996.

1.	Nonnegativity and Normalisation: Cohesion of a CBS system must be nonnegative and belong to a specified interval ([0, Coh_{max}]).
2.	Null value: Cohesion of a CBS system is null if there are no intra-component edges.
3.	Monotonicity: Addition of intra-component edges in a CBS system doesn't decrease its cohesion.
4.	Composition of components: when two components Co_1 and Co_2 are composed into a CIU (new component) $_{Co1[2,}$ the cohesion of the CBS containing the CIU $_{Co1[2}$ is not greater than that of the original CBS system containing Co_1 and Co_2.

In the remainder of this section, we will first go through the properties stated in Table 3 to show that the coupling metric Υ_D satisfies these properties. Then we will use the properties listed in Table 4 and demonstrate that our cohesion metric Ω_D satisfies the suggested properties.

2.3.1. Coupling Metric Υ_D's Properties

Since capacity of a Shannon language cannot be negative the non-negativity property, property (1) is satisfied (Table 3). When a CFG is not strongly connected, one cannot calculate the combinatorial capacity, hence property (2), the null value property is also satisfied. When we add edges between components (nodes of the CFG), the combinatorial capacity in general increases and hence the value of the Υ_D increases. Thus, the monotonicity property is satisfied. When two components are composed into a CIU that would replace the two components, the Υ_D for the overall system doesn't change as long as we do not collapse the CIU into one single node. Keeping the CIU as it was combined means that the CFG of the CBS system doesn't change. Hence property (4) is satisfied by Υ_D. Property (5) can be argued similar to the previous property. As long as one doesn't add new edges while including two components in the new CIU, the Υ_D for the overall system doesn't change. We summarise these results in Table 5.

Table 5 Υ_D versus properties listed in Table 3.

Property for a Coupling Metric	Satisfied or not
Nonnegativity	yes
Null Value	yes
Monotonicity	yes
Composition of Components	yes
Disjoint Component Additivity	yes

2.3.2. Cohesion Metric Ω_D's Properties

The first property is satisfied since the capacity of a Shannon language may not be negative (Table 4). Moreover, the maximum value for the capacity of a CFG with n nodes is given to be the capacity for the fully connected graph of the same number of nodes, denoted by C_{fc}. C_{fc} is used for normalisation. If there are no intra-component edges, one cannot calculate the capacity since strong connectivity is essential. Hence, this property is satisfied as well. When new arcs are added in a CFG representing a component, the capacity value will not decrease (C_{fc} will remain the same), Ω_D therefore possesses property (3). Our initial argument on CIUs being in-house components helps us satisfy the last property. If we have the CFG for a CIU, that indicates that we are not collapsing the internal details of the CIU. When we compose two components in a CBS system into a CIU, the Ω_D value for the CBS system will not change. This suggests that property (4) is satisfied as well. We summarise these results in Table 6.

Table 6 Ω_D versus properties listed in Table 4.

Property for a Cohesion Metric	Satisfied or not
Nonnegativity and Normalisation	yes
Null Value	yes
Monotonicity	yes
Composition of Components	yes

3. Results, Conclusions and Future Work

We presented two information-theory based metrics Ω_D and Υ_D for investigating cohesion and coupling in CBS systems. Our approach is valid at the higher levels of system design in which a CBS system is composed from readily available components. The information-theoretic modelling and metrics presented in this paper are not only valid when the system under consideration is a CBS system (where the source code is not available), but also are applicable to systems for which the source code is available.

In order to show the potential applicability of the proposed information-theoretic approach, we presented two examples and also demonstrated that the proposed metrics satisfy the sets of properties provided by (Briand *et al.*, 1996), which we adapted for CBS systems. We conclude that the information-theoretic modelling of CBS and thereby assessing coupling and cohesion early in the design phase, have the potential to aid the system architect in making critical decisions.

In this paper, for the sake of clarity and brevity, we didn't consider the possibility that a component may have more than one functionality. A component may be packed with more than one functionality in order to increase its reuse throughout various application domains. When a component has multiple functionalities, application and context dependency come into place as to which of the functionalities are used within a given application/context. This issue of multiple functionalities opens up new research directions in terms of the assessment of coupling and especially cohesion in a CBS system. We envision that our methodology can be used by utilising the "partial" CFGs for components in the system's overall CFG. However, when one refers to cohesion of a component, a clarification is needed with respect to the functionalities that are utilised and those that are not utilised. We believe, this is a legitimate problem to attack in the future studies.

4. Acknowledgment

This research was supported in part by NRF Thuthuka Programme, No. 2053850.

5. References

Allen, E.B. and Khoshgoftaar, T.M., 1999, "Measuring coupling and cohesion: an information-theory approach," Proceedings of the sixth IEEE Symposium on Software Metrics, November 4-6, pp. 119-127.

Berham, A. and Plemmons, R.J., 1979, Nonnegative Matrices in the Mathematical Sciences, Academic Press, New York.

Brereton, P. and Budgen, D., 2000, "Component-based systems: a classification of issues," IEEE Computer, Vol. 33, No. 11, pp. 54-62.

Briand, L.C., Morasca, S., and Basili, V.R., 1996, "Property-based software engineering measurement," IEEE Transactions on Software Engineering, Vol. 22, No. 1, pp. 68-86.

Brown, A.W. and Wallnau, K.C., 1998, "The current state of CBSE," IEEE Software, pp. 37-46.

Chaitin, G.J., 1975, Randomness and mathematical proof, Scientific American, Vol. 232, No. 5, pp. 47-52.

Coulter, N., Cooper, R.B., and Solomon, M.K., 1987, "Information-theoretic complexity of program specifications," The Computer Journal, Vol. 30, No. 3, pp. 223-227.

Cox, P.T. and Baoming, S., 2001, "A formal model for component-based software," IEEE Symposia on Human-Centric Computing Languages and Environments, pp. 304-311.

Emmerich, W., 2002, "Distributed component technologies and their software engineering implications," Proceedings of the 24th International Conference on Software Engineering, Orlando, FL, pp. 537-546.

Grundy, J., Mugridge, R., Hosking, J., and Apperley, M., 1998, “Tool integration, collaboration and user interaction issues in component-based software architectures,” Technology of Object-Oriented Languages, pp. 299-312.

Heineman, G.T. and Councill, W.T., 2001, Component-Based Software Engineering: Putting the Pieces Together, Addison-Wesley, Massachusetts.

Hellerman, L., 1972, “A measure of computational work,” IEEE Transactions on Computers, Vol. C-21, No. 5, pp. 439-446.

Hopkins, J., 2000, “Component primer,” Communications of the ACM, Vol. 43, No. 10, pp. 27–30.

Jololian, L.K., 2000, A Meta-Semantic Language for Smart Component-Adapters, PhD thesis, New Jersey Institute of Technology, New Jersey.

Khandekar, A., McEliece, R., and Rodemich, E., 1999, “The discrete noiseless channel revisited,” Procceeding of 1999 International Symposium Comm, Theory and Applications, pp. 115-137.

Lind, D. and Markus, B., 1999, An Introduction to Symbolic Dynamics and Coding, Cambridge Univ. Press.

Martins, E., Toyota, C.M., and Yanagawa, R.L., 2001, “Constructing self-testable software components,” The International Conference on Dependable Systems and Networks, pp, 151-160.

Minc, H., 1988, Nonnegative Matrices, John Wiley and Sons, New York.

Muchnick, S.S. and Jones, N.D., 1981, Program Flow Analysis: Theory and Applications, Prentice-Hall, Inc.

Nambiar, K.K., 1996, “Matrices with elements from a division ring,” Mathematical and Computer Modelling, Vol. 24, No. 1, pp. 1-3.

Nambiar, K.K., 2001, “Shannon's communication channels and word spaces,” Mathematical and Computer Modelling, Vol. 34, No. 7-8, pp. 757-759.

Ramamoorthy, C.V., 1966, Analysis of graphs by connectivity considerations, Journal of the ACM, Vol. 13, No. 2, pp. 211-222.

Ramamoorthy, C.V. and Ho, S.-B.F., 1975, “Testing large software with automated software evaluation systems,” IEEE Transactions on Software Engineering, Vol. SE-1, No. 1, pp. 46-58.

Ramamoorthy, C.V., Ho, S.-B.F., and Chen, W.T., 1976, “On the automated generation of program test data,” IEEE Transactions on Software Engineering, Vol. SE-2, No. 4, pp. 293-300.

Sage, A.P. and Palmer, J.D., 1990, Software Systems Engineering, John Wiley & Sons.

Schutt, D., 1977, “On a hypergraph oriented measure for applied computer science,” Proceedings of COMPCON.

Seker, R. and Tanik, M.M., 2001, Division ring, information theory, and discrete noiseless communication channel, Technical Report 2001-12-ECE-018, University of Alabama at Birmingham.

Seker, R. and Tanik, M.M., 2002, Discrete noiseless information channel and search engines, Technical Report 2002-04-ECE-002, University of Alabama at Birmingham.

Seker, R., 2002, Component-Based Software Modeling Based on Shannon's Information Channels, PhD thesis, University of Alabama at Birmingham.

Seker, R., Aktunc, O., Ozaydin, B., Jololian, L.K., and Tanik, M.M., 2004, “Pervasive shannon metrics and component-based software,” International Journal of Computer and Information Science, Vol. 5, No. 1. (to appear).

Seker, R. and Tanik, M.M., 2003a, “Component-based software modeling based on noisy information channels,” in: H.-W. Chu, J. Ferrer, and J.M. Pineda, editors, Proceedings of the International Conference on Computer, Communication, and Control Technologies(CCCT'03), Volume 3, Orlando, FL, July 31-August 2, pp. 144-149.

Seker, R. and Tanik, M.M., 2003b, "Component-based software modeling based on Shannon languages," in: R.Y. Lee and K.W. Lee, editors, Proceedings of the International Conference on Software Engineering Research and Applications (ICSERA'03), San Francisco, CA, June 25-27, pp. 185-191.

Shannon, C.E. and Weaver, W., 1963, The Mathematical Theory of Communication, University of Illinois Press, Urbana.

Tanik, M.M. and Chan, E.S., 1991, Fundamentals of Computing for Software Engineers, Van Nostrand Reinhold, New York, NY.

van den Broek, P.M. and van den Berg, K.G., 1995, "Generalised approach to software structure metrics," Software Engineering Journal, Vol. 10, No. 2, pp. 61-67.

Wallnau, K.C., Hissam, S.A., and Seacord, R.C., 2002, Building Systems from Commercial Components, Addison-Wesley, Massachusetts.